Heterogeneous Gaussian Mechanism: Preserving Differential Privacy in Deep Learning with Provable Robustness

NhatHai Phan[1*] , **Minh N. Vu**[5*] , **Yang Liu**[1*] ,
Ruoming Jin[2] , **Dejing Dou**[3] , **Xintao Wu**[4] and **My T. Thai**[5]
[1]New Jersey Institute of Technology, Newark, New Jersey, USA
[2]Kent State University, Kent, Ohio, USA
[3]University of Oregon, Eugene, Oregon, USA
[4]University of Arkansas, Fayetteville, Arkansas, USA
[5]University of Florida, Gainesville, Florida, USA
{phan, yl558}@njit.edu, {minhvu, mythai}@ufl.edu, rjin1@kent.edu, dou@uoregon.edu,
xintaowu@uark.edu

Abstract

In this paper, we propose a novel Heterogeneous Gaussian Mechanism (HGM) to preserve differential privacy in deep neural networks, with provable robustness against adversarial examples. We first relax the constraint of the privacy budget in the traditional Gaussian Mechanism from (0, 1] to (0, infty), with a new bound of the noise scale to preserve differential privacy. The noise in our mechanism can be arbitrarily redistributed, offering a distinctive ability to address the trade-off between model utility and privacy loss. To derive provable robustness, our HGM is applied to inject Gaussian noise into the first hidden layer. Then, a tighter robustness bound is proposed. Theoretical analysis and thorough evaluations show that our mechanism notably improves the robustness of differentially private deep neural networks, compared with baseline approaches, under a variety of model attacks.

1 Introduction

Recent developments of machine learning (ML) significantly enhance sharing and deploying of ML models in practical applications more than ever before. This presents critical privacy and security issues, when ML models are built on personal data, e.g., clinical records, images, user profiles, etc. In fact, adversaries can conduct: 1) privacy model attacks, in which deployed ML models can be used to reveal sensitive information in the private training data [Fredrikson *et al.*, 2015; Wang *et al.*, 2015; Shokri *et al.*, 2017; Papernot *et al.*, 2016]; and 2) adversarial example attacks [Goodfellow *et al.*, 2014] to cause the models to misclassify. Note that adversarial examples are maliciously perturbed inputs designed to mislead a model at test time [Liu *et al.*, 2016; Carlini and Wagner, 2017]. That poses serious risks to deploy machine learning models in practice. Therefore, it is of paramount significance to simultaneously preserve privacy in the private training data and guarantee the robustness of the model under adversarial examples.

To preserve privacy in the training set, recent efforts have focused on applying Gaussian Mechanism (**GM**) [Dwork and Roth, 2014] to preserve differential privacy (**DP**) in deep learning [Abadi *et al.*, 2016; Hamm *et al.*, 2017; Yu *et al.*, 2019; Lee and Kifer, 2018]. The concept of DP is an elegant formulation of privacy in probabilistic terms, and provides a rigorous protection for an algorithm to avoid leaking personal information contained in its inputs. It is becoming mainstream in many research communities and has been deployed in practice in the private sector and government agencies. DP ensures that the adversary cannot infer any information with high confidence (controlled by a privacy budget ϵ and a broken probability δ) about any specific tuple from the released results. GM is also applied to derive provable robustness against adversarial examples [Lecuyer *et al.*, 2018]. However, existing efforts only focus on either preserving DP or deriving provable robustness [Kolter and Wong, 2017; Raghunathan *et al.*, 2018], but not both DP and robustness!

With the current form of GM [Dwork and Roth, 2014] applied in existing works [Abadi *et al.*, 2016; Hamm *et al.*, 2017; Lecuyer *et al.*, 2018], it is challenging to preserve DP in order to protect the training data, with provable robustness. In GM, random noise scaled to $\mathcal{N}(0, \sigma^2)$ is injected into each of the components of an algorithm output, where the noise scale σ is a function of ϵ, δ, and the mechanism sensitivity Δ. There are three major limitations in these works when applying GM: **(1)** The privacy budget ϵ in GM is restricted to $(0, 1]$, resulting in a limited search space to optimize the model utility and robustness bounds; **(2)** All the features (components) are treated the same in terms of the amount of noise injected. That may not be optimal in real-world scenarios [Bach *et al.*, 2015; Phan *et al.*, 2017]; and **(3)** Existing works have not been designed to defend against adversarial examples, while preserving DP in order to protect the training data. These limitations do narrow the applicability of GM, DP, deep learning, and provable robustness, by affecting the model utility, flexibility, reliability, and resilience to model attacks in practice.

[*] Co-first authors.

Our Contributions

To address these issues, we first propose a novel *Heterogeneous Gaussian Mechanism* (**HGM**), in which **(1)** the constraint of ϵ is extended from $(0, 1]$ to $(0, \infty)$; **(2)** a new lower bound of the noise scale σ will be presented; and more importantly, **(3)** the magnitude of noise can be heterogeneously injected into each of the features or components. These significant extensions offer a distinctive ability to address the trade-off among model utility, privacy loss, and robustness by redistributing the noise and enlarging the search space for better defensive solutions.

Second, we develop a novel approach, called **Secure-SGD**, to achieve both DP and robustness in the general scenario, i.e., any value of the privacy budget ϵ. In Secure-SGD, our HGM is applied to inject Gaussian noise into the first hidden layer of a deep neural network. This noise is used to derive *a tighter and provable robustness bound.* Then, DP stochastic gradient descent (**DPSGD**) algorithm [Abadi *et al.*, 2016] is applied to learn differentially private model parameters. The training process of our mechanism preserves DP in deep neural networks to protect the training data with provable robustness. To our knowledge, Secure-SGD is the first approach to learn such a secure model with a high utility. Rigorous experiments conducted on MNIST and CIFAR-10 datasets [Lecun *et al.*, 1998; Krizhevsky and Hinton, 2009] show that our approach significantly improves the robustness of DP deep neural networks, compared with baseline approaches.

2 Preliminaries and Related Work

In this section, we revisit differential privacy, PixelDP [Lecuyer *et al.*, 2018], and introduce our problem definition. Let D be a database that contains n tuples, each of which contains data $x \in [-1, 1]^d$ and a *ground-truth label* $y \in \mathbb{Z}_K$. Let us consider a classification task with K possible categorical outcomes; i.e., the data label y given $x \in D$ is assigned to only one of the K categories. Each y can be considered as a one-hot vector of K categories $y = \{y_1, \ldots, y_K\}$. On input x and parameters θ, a model outputs class scores $f : \mathbb{R}^d \to \mathbb{R}^K$ that maps d-dimentional inputs x to a vector of scores $f(x) = \{f_1(x), \ldots, f_K(x)\}$ s.t. $\forall k : f_k(x) \in [0, 1]$ and $\sum_{k=1}^{K} f_k(x) = 1$. The class with the highest score value is selected as the *predicted label* for the data tuple, denoted as $y(x) = \max_{k \in K} f_k(x)$. We specify a loss function $L(f(x), y)$ that represents the penalty for mismatching between the predicted values $f(x)$ and original values y.

Differential Privacy

The definitions of differential privacy and Gaussian Mechanism are as follows:

Definition 1 (ϵ, δ)*-Differential Privacy [Dwork* et al.*, 2006]. A randomized algorithm A fulfills (ϵ, δ)-differential privacy, if for any two databases D and D' differing at most one tuple, and for all $\mathbf{o} \subseteq Range(A)$, we have:*

$$Pr[A(D) = \mathbf{o}] \leq e^{\epsilon} Pr[A(D') = \mathbf{o}] + \delta \tag{1}$$

Smaller ϵ and δ enforce a stronger privacy guarantee.

Here, ϵ controls the amount by which the distributions induced by D and D' may differ, and δ is a broken probability. DP also applies to general metrics $\rho(D, D') \leq 1$, including Hamming metric as in Definition 1 and $l_{p\in\{1,2,\infty\}}$-norms [Chatzikokolakis *et al.*, 2013]. Gaussian Mechanism is applied to achieve DP given a random algorithm A as follows:

Theorem 1 *Gaussian Mechanism [Dwork and Roth, 2014]. Let $A : \mathbb{R}^d \to \mathbb{R}^K$ be an arbitrary K-dimensional function, and define its l_2 sensitivity to be $\Delta_A = \max_{D,D'} \|A(D) - A(D')\|_2$. The Gaussian Mechanism with parameter σ adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to each of the K components of the output. Given $\epsilon \in (0, 1]$, the Gaussian Mechanism with $\sigma \geq \sqrt{2\ln(1.25/\delta)}\Delta_A/\epsilon$ is (ϵ, δ)-DP.*

Adversarial Examples

For some target model f and inputs (x, y_{true}), i.e., y_{true} is the true label of x, one of the adversary's goals is to find an *adversarial example* $x^{\text{adv}} = x + \alpha$, where α is the perturbation introduced by the attacker, such that: **(1)** x^{adv} and x are close, and **(2)** the model misclassifies x^{adv}, i.e., $y(x^{\text{adv}}) \neq y(x)$. In this paper, we consider well-known classes of $l_{p\in\{1,2,\infty\}}$-norm bounded attacks [Goodfellow *et al.*, 2014]. Let $l_p(\mu) = \{\alpha \in \mathbb{R}^d : \|\alpha\|_p \leq \mu\}$ be the l_p-norm ball of radius μ. One of the goals in adversarial learning is to minimize the risk over adversarial examples:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y_{\text{true}})\sim\mathcal{D}} \Big[\max_{\|\alpha\|_p \leq \mu} L\big(f(x + \alpha, \theta), y_{\text{true}}\big) \Big]$$

where a specific attack is used to approximate solutions to the inner maximization problem, and the outer minimization problem corresponds to training the model f with parameters θ over these adversarial examples $x^{\text{adv}} = x + \alpha$.

We revisit two basic attacks in this paper. The first one is a *single-step* algorithm, in which only a single gradient computation is required. For instance, Fast Gradient Sign Method (**FGSM**) algorithm [Goodfellow *et al.*, 2014] finds an adversarial example by maximizing the loss function $L(f(x^{\text{adv}}, \theta), y_{\text{true}})$. The second one is an *iterative* algorithm, in which multiple gradients are computed and updated. For instance, in [Kurakin *et al.*, 2016], FGSM is applied multiple times with small steps, each of which has a size of μ/T_μ, where T_μ is the number of steps.

Provable Robustness and PixelDP

In this paper, we consider the following robustness definition. Given a benign example x, we focus on achieving a robustness condition to attacks of $l_p(\mu)$-norm, as follows:

$$\forall \alpha \in l_p(\mu) : f_k(x + \alpha) > \max_{i:i\neq k} f_i(x + \alpha) \tag{2}$$

where $k = y(x)$, indicating that a small perturbation α in the input does not change the predicted label $y(x)$.

To achieve the robustness condition in Eq. 2, [Lecuyer *et al.*, 2018] introduce an algorithm, called **PixelDP**. By considering an input x (e.g., images) as databases in DP parlance, and individual features (e.g., pixels) as tuples in DP, PixelDP shows that randomizing the scoring function $f(x)$ to enforce DP on a small number of pixels in an image guarantees robustness of predictions against adversarial examples that can change up to that number of pixels. To achieve the goal, noise $\mathcal{N}(0, \sigma_r^2)$ is injected into either input x or some hidden layer of a deep neural network. That results in the following (ϵ_r, δ_r)-PixelDP condition, with a budget ϵ_r and a broken brobability δ_r of robustness, as follows:

Lemma 1 *(ϵ_r, δ_r)-PixelDP [Lecuyer* et al.*, 2018]. Given a randomized scoring function $f(x)$ satisfying (ϵ_r, δ_r)-PixelDP w.r.t. a l_p-norm metric, we have:*

$$\forall k, \forall \alpha \in l_p(\mu = 1) : \mathbb{E}f_k(x) \leq e^{\epsilon_r}\mathbb{E}f_k(x + \alpha) + \delta_r \quad (3)$$

where $\mathbb{E}f_k(x)$ is the expected value of $f_k(x)$.

The network is trained by applying typical optimizers, such as SGD. At the prediction time, a certified robustness check is implemented for each prediction. A generalized robustness condition is proposed as follows:

$$\forall \alpha \in l_p(\mu = 1) : \hat{\mathbb{E}}_{lb}f_k(x) > e^{2\epsilon_r} \max_{i:i \neq k} \hat{\mathbb{E}}_{ub}f_i(x) + (1 + e^{\epsilon_r})\delta_r \quad (4)$$

where $\hat{\mathbb{E}}_{lb}$ and $\hat{\mathbb{E}}_{ub}$ are the lower bound and upper bound of the expected value $\hat{\mathbb{E}}f(x) = \frac{1}{N}\sum_N f(x)_N$, derived from the Monte Carlo estimation with an η-confidence, given N is the number of invocations of $f(x)$ with independent draws in the noise σ_r. Passing the check for a given input x guarantees that *no perturbation exists up to $l_p(\mu = 1)$-norm that causes the model to change its prediction result*. In other words, the classification model, based on $\hat{\mathbb{E}}f(x)$, i.e., $\arg\max_k \hat{\mathbb{E}}f_k(x)$, is consistent to attacks of $l_p(\mu = 1)$-norm on x with probability $\geq \eta$. Group privacy [Dwork *et al.*, 2006] can be applied to achieve the same robustness condition, given a particular size of perturbation $l_p(\mu)$. For a given σ_r, δ_r, and sensitivity $\Delta_{p,2}$ used at prediction time, PixelDP solves for the maximum μ for which the robustness condition in Eq. 4 checks out:

$$\boxed{\begin{aligned} &\mu_{max} = \max_{\mu \in \mathbb{R}^+} \mu \quad \text{such that} \quad \forall \alpha \in l_p(\mu) : \\ &\hat{\mathbb{E}}_{lb}f_k(x) > e^{2\epsilon_r} \max_{i:i \neq k} \hat{\mathbb{E}}_{ub}f_i(x) + (1 + e^{\epsilon_r})\delta_r \\ &\sigma_r = \sqrt{2\ln(1.25/\delta_r)}\Delta_{p,2}\mu/\epsilon_r \quad \text{and} \quad \epsilon_r \leq 1 \end{aligned}} \quad (5)$$

3 Heterogeneous Gaussian Mechanism

We now formally present our Heterogeneous Gaussian Mechanism (HGM) and the Secure-SGD algorithm. In Eq. 5, it is clear that ϵ is restricted to be $(0, 1]$, following the Gaussian Mechanism (Theorem 1). That affects the robustness bound in terms of flexibility, reliability, and utility. In fact, adversaries only need to guarantee that $\hat{\mathbb{E}}_{lb}f_k(x + \alpha)$ is larger than at most $e^2 \max_{i:i \neq k} \hat{\mathbb{E}}_{ub}f_i(x + \alpha) + (1 + e)\delta$, i.e., $\epsilon_r = 1$, in order to assault the robustness condition: thus, softening the robustness bound. In addition, the search space for the robustness bound μ_{max} is limited, given $\epsilon \in (0, 1]$. These issues increase the number of robustness violations, potentially degrading the utility and reliability of the robustness bound. In real-world applications, such as healthcare, autonomous driving, object recognition, etc., a flexible value of ϵ_r is needed to implement stronger and more practical robustness bounds. This is also true for many other algorithms applying Gaussian Mechanism [Dwork and Roth, 2014].

To relax this constraint, we introduce an Extended Gaussian Mechanism as follows:

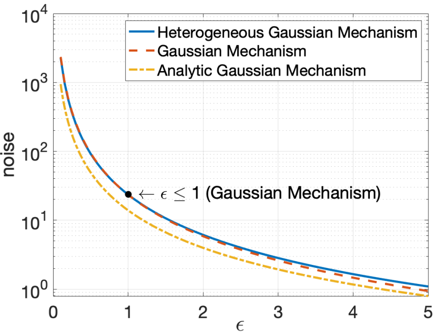


Figure 1: The magnitude of Gaussian noise, given the traditional Gaussian Mechanism, Analytic Gaussian Mechanism, and our Heterogeneous Gaussian Mechanism.

Theorem 2 *Extended Gaussian Mechanism. Let $A : \mathbb{R}^d \rightarrow \mathbb{R}^K$ be an arbitrary K-dimensional function, and define its l_2 sensitivity to be $\Delta_A = \max_{D,D'} \|A(D) - A(D')\|_2$. An Extended Gaussian Mechanism M with parameter σ adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to each of the K components of the output. The mechanism M is (ϵ, δ)-DP, with*

$$\epsilon > 0, \quad \sigma \geq \frac{\sqrt{2}\Delta_A}{2\epsilon}(\sqrt{s} + \sqrt{s + \epsilon}), \textit{ and } \quad s = \ln(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})$$

Detailed proof of Theorem 2 is in **Appendix B**[1]. The Extended Gaussian Mechanism enables us to relax the constraint of ϵ. However, the noise scale σ is used to inject Gaussian noise into each component. This may not be optimal, since different components usually have different impacts to the model outcomes [Bach *et al.*, 2015]. To address this, we further propose a Heterogeneous Gaussian Mechanism (HGM), in which the noise scale σ in Theorem 2 can be arbitrarily redistributed. Different strategies can be applied to improve the model utility and to enrich the search space for better robustness bounds. For instance, *more noise* will be injected into *less important* components, or vice-versa, or even randomly redistributed. In order to achieve our goal, we introduce a noise redistribution vector $K\mathbf{r}$, where $\mathbf{r} \in \mathbb{R}^K$ that satisfies $0 \leq r_i \leq 1$ $(i \in [K])$ and $\sum_{i=1}^K r_i = 1$. We show that by injecting Gaussian noise $\mathcal{N}(0, \sigma^2 K\mathbf{r})$, where $\Delta_A = \max_{D,D'} \sqrt{\sum_{k=1}^K \frac{1}{Kr_k}\big(A(D)_k - A(D')_k\big)^2}$ and $\rho(D, D') \leq 1$, we achieve (ϵ, δ)-DP.

Theorem 3 *Heterogeneous Gaussian Mechanism. Let $A : \mathbb{R}^d \rightarrow \mathbb{R}^K$ be an arbitrary K-dimensional function, and define its l_2 sensitivity to be $\Delta_A = \max_{D,D'} \|\frac{A(D) - A(D')}{\sqrt{K\mathbf{r}}}\|_2 = \max_{D,D'} \sqrt{\sum_{k=1}^K \frac{1}{Kr_k}\big(A(D)_k - A(D')_k\big)^2}$. A Heterogeneous Gaussian Mechanism M with parameter σ adds noise scaled to $\mathcal{N}(0, \sigma^2 K\mathbf{r})$ to each of the K components of the output. The mechanism M is (ϵ, δ)-DP, with*

$$\epsilon > 0, \quad \sigma \geq \frac{\sqrt{2}\Delta_A}{2\epsilon}(\sqrt{s} + \sqrt{s + \epsilon}), \textit{ and } \quad s = \ln(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})$$

where $\mathbf{r} \in \mathbb{R}^K$ s.t. $0 \leq r_i \leq 1$ $(i \in [K])$ and $\sum_{i=1}^K r_i = 1$.

[1] https://www.dropbox.com/s/mjkq4zqqh6ifqir/HGM_Appendix.pdf?dl=0

Detailed proof of Theorem 3 is in **Appendix C**[1]. It is clear that the Extended Gaussian Mechanism is a special case of the HGM, when $\forall i \in [K] : r_i = 1/K$. Figure 1 illustrates the magnitude of noise injected by the traditional Gaussian Mechanism, the state-of-the-art Analytic Gaussian Mechanism [Balle and Wang, 2018], and our Heterogeneous Gaussian Mechanism as a function of ϵ, given the global sensitivity $\Delta_A = 1$, and $\delta = 1e{-}5$ (a very tight broken probability), and $\forall i \in [K] : r_i = 1/K$. The lower bound of the noise scale in our HGM is just a little bit better than the traditional Gaussian Mechanism when $\epsilon \leq 1$. However, our mechanism does not have the constraint $(0, 1]$ on the privacy budget ϵ. The Analytic Gaussian Mechanism, which provides the state-of-the-art noise bound, has a better noise scale than our mechanism. However, our noise scale bound provides a distinctive ability to redistribute the noise via the vector $K\mathbf{r}$, compared with the Analytic Gaussian Mechanism. There could be numerous strategies to identify vector $\mathbf{r}$. This is significant when addressing the trade-off between model utility and privacy loss or robustness in real-world applications. In our mechanism, *"more noise"* is injected into *"more vulnerable"* components to improve the robustness. We will show how to compute vector $\mathbf{r}$ and identify vulnerable components in our Secure-SGD algorithm. Experimental results illustrate that, by redistributing the noise, our HGM yields better robustness, compared with existing mechanisms.

4 Secure-SGD

In this section, we focus on applying our HGM in a crucial and emergent application, which is enhancing the robustness of differentially private deep neural networks. Given a deep neural network f, DPSGD algorithm [Abadi *et al.*, 2016] is applied to learn (ϵ, δ)-DP parameters θ. Then, by injecting Gaussian noise into the first hidden layer, we can leverage the robustness concept of PixelDP [Lecuyer *et al.*, 2018] (Eq. 5) to derive a better robustness bound based on our HGM.

Algorithm 1 (**Appendix A**[1]) outlines the key steps in our Secure-SGD algorithm. We first initiate the parameters θ and construct a deep neural network $f : \mathbb{R}^d \to \mathbb{R}^K$ (*Lines 1-2*). Then, a robustness noise $\gamma \leftarrow \mathcal{N}(0, \sigma_r^2 K\mathbf{r})$ is drawn by applying our HGM (*Line 3*), where σ_r is computed following Theorem 3, K is the number of hidden neurons in h_1, denoted as $K = |h_1|$, and Δ_f is the sensitivity of the algorithm, defined as the maximum change in the output (i.e., which is $h_1(x) = W_1^T x$) that can be generated by the perturbation in the input x under the noise redistribution vector $K\mathbf{r}$.

$$\Delta_f = \max_{x, x' : x \neq x'} \frac{\| \frac{h_1(x) - h_1(x')}{\sqrt{K\mathbf{r}}} \|_2}{\|x - x'\|_\infty} \leq \| \frac{W_1}{\sqrt{K\mathbf{r}}} \|_{\infty,2} \quad (6)$$

For l_∞-norm attacks, we use the following bound $\Delta_f = \sqrt{|h_1|} \| \frac{W_1}{K\mathbf{r}} \|_\infty$, where $\| \frac{W_1}{K\mathbf{r}} \|_\infty$ is the maximum 1-norm of W_1's rows over the vector $K\mathbf{r}$. The vector $\mathbf{r}$ can be computed as the forward derivative of $h_1(x)$ as follows:

$$\mathbf{r} = \frac{\mathbf{s}}{\sum_{s_i \in \mathbf{s}} s_i}, \textit{ where } \mathbf{s} = \frac{1}{n} \sum_{x \in D} \Big| \frac{\partial L(\theta, x)}{\partial h_1(x)} \Big|^\beta \quad (7)$$

where β is a user-predefined inflation rate. It is clear that features, which have higher forward derivative values, will be more vulnerable to attacks by maximizing the loss function $L(\theta, x)$. These features are assigned larger values in vector $\mathbf{r}$, resulting in more noise injected, and vice-versa. The computation of $\mathbf{r}$ can be considered as a prepossessing step using a pre-trained model. It is important to note that the utilizing of $\mathbf{r}$ does not risk any privacy leakage, since $\mathbf{r}$ is only applied to derive provable robustness. It does not have any effect on the DP-preserving procedure in our algorithm, as follows. First, at each training step $t \in T$, our mechanism takes a random sample B_t from the data D, with sampling probability m/n, where m is a batch size (*Line 5*). For each tuple $x_i \in B_t$, the first hidden layer is perturbed by adding Gaussian noise derived from our HGM (*Line 6, Alg. 1*):

$$h_1(x_i) = W_1^T x_i + \gamma \quad (8)$$

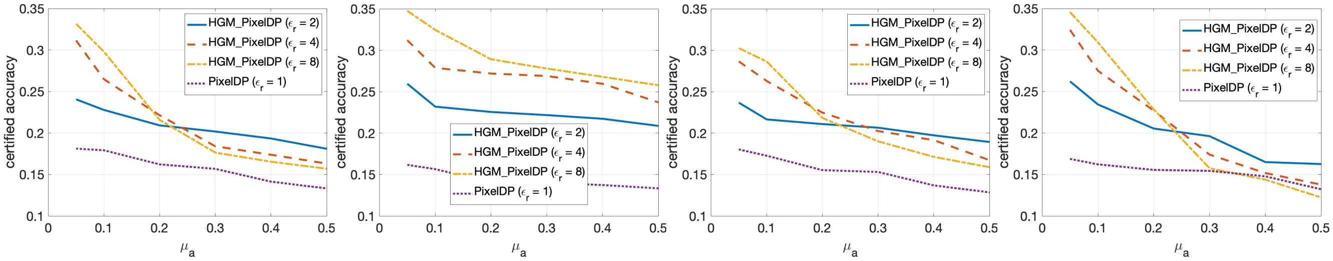
This ensures that the scoring function $f(x)$ satisfies (ϵ_r, δ_r)-PixelDP (Lemma 3). Then, the gradient $\mathbf{g}_t(x_i) = \nabla_{\theta_t} L(\boldsymbol{\theta}_t, x_i)$ is computed (*Lines 7-9*). The gradients will be bounded by clipping each gradient in l_2 norm; i.e., the gradient vector $\mathbf{g}_t(x_i)$ is replaced by $\mathbf{g}_t(x_i) / \max(1, \|\mathbf{g}_t(x_i)\|_2 / C)$ for a predefined threshold C (*Lines 10-12*). Uniformed normal distribution noise is added into gradients of parameters $\boldsymbol{\theta}$ (*Line 14*), as:

$$\widetilde{g}_t \leftarrow \frac{1}{m} \Big(\sum_i \frac{\mathbf{g}_t(x_i)}{\max(1, \frac{\|\mathbf{g}_t(x_i)^2\|}{C})} + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \Big) \quad (9)$$

The descent of the parameters explicitly is as: $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \xi_t \widetilde{g}_t$, where ξ_t is a learning rate at the step t (*Line 16*). The training process of our mechanism achieves both (ϵ, δ)-DP to protect the training data and provable robustness with the budgets (ϵ_r, δ_r). In the verified testing phase (*Lines 17-22*), by applying HGM and PixelDP, we derive a novel robustness bound μ_{max} for a specific input x as follows:

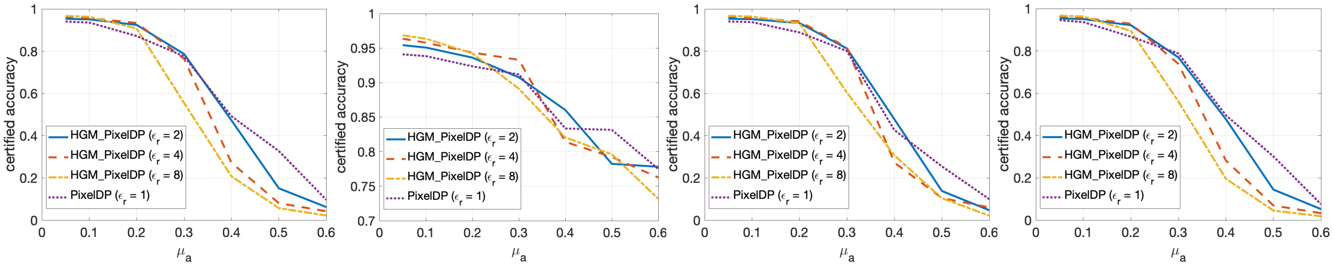
$$\boxed{\begin{aligned} &\mu_{max} = \max_{\mu \in \mathbb{R}^+} \mu, \quad \text{such that} \quad \forall \alpha \in l_p(\mu) : \\ &\hat{\mathbb{E}}_{lb} f_k(x) > e^{2\epsilon_r} \max_{i : i \neq k} \hat{\mathbb{E}}_{ub} f_i(x) + (1 + e^{\epsilon_r}) \delta_r \\ &\sigma_r = \frac{\sqrt{2}}{2\epsilon_r} (\sqrt{s} + \sqrt{s + \epsilon_r}) \Delta_f \times \mu / \epsilon_r \ \text{ and } \ \epsilon_r > 0 \end{aligned}} \quad (10)$$

where $\hat{\mathbb{E}}_{lb}$ and $\hat{\mathbb{E}}_{ub}$ are the lower and upper bounds of the expected value $\hat{\mathbb{E}} f(x) = \frac{1}{N} \sum_N f(x)_N$, derived from the Monte Carlo estimation with an η-confidence, given N is the number of invocations of $f(x)$ with independent draws in the noise $\gamma \leftarrow \mathcal{N}(0, \sigma_r^2 K\mathbf{r})$. Similar to [Lecuyer *et al.*, 2018], we use Hoeffding's inequality [Hoeffding, 1963] to bound the error in $\hat{\mathbb{E}} f(x)$. If the robustness size μ_{max} is larger than a given adversarial perturbation size μ_a, the model prediction is considered consistent to that attack size. Given the relaxed budget $\epsilon_r > 0$ and the noise redistribution $K\mathbf{r}$, the search space for the robustness size μ_{max} is significantly enriched, e.g., $\epsilon_r > 1$, strengthening the robustness bound. Note that vector $\mathbf{r}$ can also be randomly drawn in the estimation of the expected value $\hat{\mathbb{E}} f(x)$. Both fully-connected and convolution layers can be applied. Given a convolution layer, we need to ensure that the computation of each feature map is (ϵ_r, δ_r)-PixelDP, since each of them is independently computed by



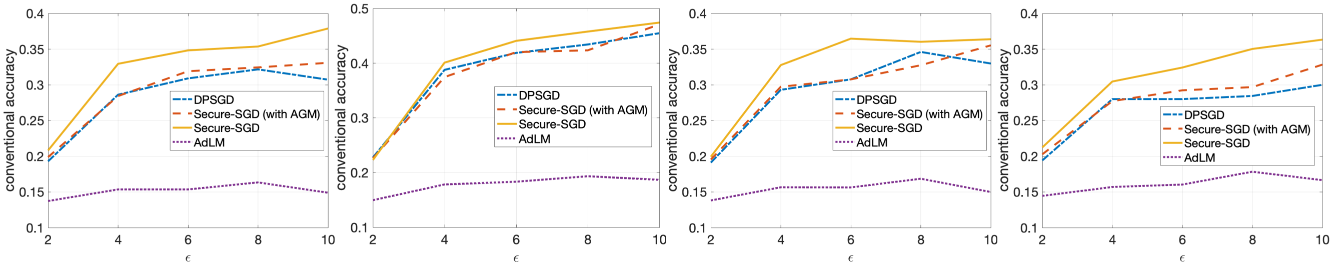
(a) I-FGSM attacks (b) FGSM attacks (c) MIM attacks (d) MadryEtAl attacks

Figure 2: Certified accuracy on the CIFAR-10 dataset, given HGM_PixelDP and PixelDP (i.e., no DP preservation).



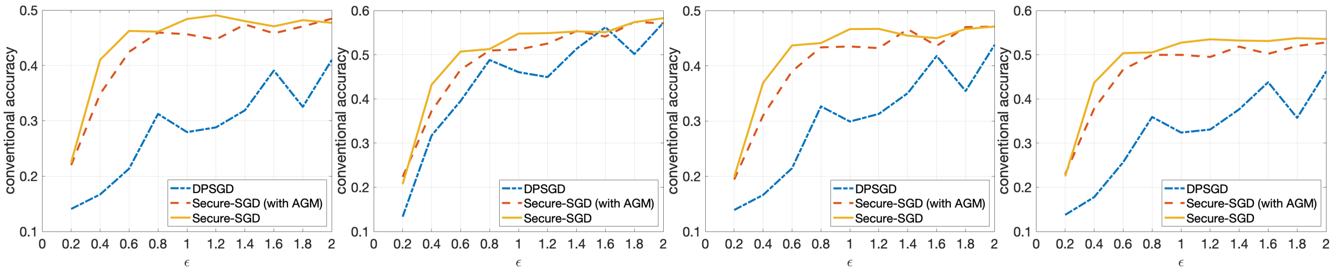
(a) I-FGSM attacks (b) FGSM attacks (c) MIM attacks (d) MadryEtAl attacks

Figure 3: Certified accuracy on the MNIST dataset, given HGM_PixelDP and PixelDP (i.e., no DP preservation).



(a) I-FGSM attacks (b) FGSM attacks (c) MIM attacks (d) MadryEtAl attacks

Figure 4: Conventional accuracy on the CIFAR-10 dataset, given Secure-SGD, DPSGD, and AdLM, i.e., $l_\infty(\mu_a = 0.2)$, $\epsilon_r = 8$.



(a) I-FGSM attacks (b) FGSM attacks (c) MIM attacks (d) MadryEtAl attacks

Figure 5: Conventional accuracy on the MNIST dataset, given Secure-SGD and DPSGD, i.e., $l_\infty(\mu_a = 0.1)$, $\epsilon_r = 4$.

reading a local region of input neurons. Therefore, the sensitivity Δ_f can be considered the upper-bound sensitivity given any single feature map. Our algorithm is the first effort to connect DP preservation in order to protect the original training data and provable robustness in deep learning.

5 Experimental Results

We have carried out extensive experiments on two benchmark datasets, MNIST and CIFAR-10. Our goal is to evaluate whether our HGM significantly improves the robustness of both differentially private and non-private models under strong adversarial attacks, and whether our Secure-SGD approach retains better model utility compared with baseline mechanisms, under the same DP guarantees and protections.

Baseline approaches. Our **HGM** and two approaches, including **HGM_PixelDP** and **Secure-SGD**, are evaluated in comparison with state-of-the-art mechanisms in: (1) DP-preserving algorithms in deep learning, i.e., **DPSGD** [Abadi *et al.*, 2016], **AdLM** [Phan *et al.*, 2017]; in (2) Provable robustness, i.e., **PixelDP** [Lecuyer *et al.*, 2018]; and (3) The Analytic Gaussian Mechanism (**AGM**) [Balle and Wang, 2018]. To preserve DP, DPSGD injects random noise into gradients of parameters, while AdLM is a Functional Mechanism-based approach. PixelDP is one of the state-of-the-art mechanisms providing provable robustness using DP bounds. Our **HGM_PixelDP** model simply is PixelDP with the noise bound derived from our HGM. The baseline models share the same design in our experiment. We consider the class of l_∞-bounded adversaries. Four white-box attack algorithms were used, including **FGSM**, **I-FGSM**, Momentum Iterative Method (**MIM**) [Dong *et al.*, 2017], and **MadryEtAl** [Madry *et al.*, 2018], to draft adversarial examples $l_\infty(\mu_a)$.

Datasets and configurations. *MNIST:* We used two convolution layers (32 and 64 features). Each hidden neuron connects with a 5x5 unit patch. A fully-connected layer has 256 units. The batch size m was set to 128, $\xi = 1.5$, $\psi = 2$, $T_\mu = 10$, and $\beta = 1$. *CIFAR-10:* We used three convolution layers (128, 128, and 256 features). Each hidden neuron connects with a 3x3 unit patch in the first layer, and a 5x5 unit patch in other layers. One fully-connected layer has 256 neurons. The batch size m was set to 128, $\xi = 1.5$, $\psi = 10$, $T_\mu = 3$, and $\beta = 1$. Note that $\boldsymbol{\epsilon}$ is used to indicate the DP budget used to protect the training data; meanwhile, $\boldsymbol{\epsilon_r}$ is the budget for robustness. The implementation of our mechanism is available in TensorFlow[2]. We apply two accuracy metrics as follows:

$$\textit{conventional accuracy} = \frac{\sum_{i=1}^{|test|} isCorrect(x_i)}{|test|}$$

$$\textit{certified accuracy} = \frac{\sum_{i=1}^{|test|} isCorrect(x_i) \ \& \ isRobust(x_i)}{|test|}$$

where $|test|$ is the number of test cases, $isCorrect(\cdot)$ returns 1 if the model makes a correct prediction (otherwise, returns 0), and $isRobust(\cdot)$ returns 1 if the robustness size is larger than a given attack bound μ_a (otherwise, returns 0).

HGM_PixelDP. Figures 2 and 3 illustrate the certified accuracy under attacks of each model as a function of the adversarial perturbation μ_a. Our HGM_PixelDP notably outperforms the PixelDP model in most of the cases given the CIFAR-10 dataset. We register an improvement of 8.63% on average when $\epsilon_r = 8$ compared with the PixelDP, i.e., $p < 8.14e - 7$ (2 tail t-test). This clearly shows the effectiveness of our HGM in enhancing the robustness against adversarial examples. Regarding the MNIST data, our HGM_PixelDP model achieves better certified accuracies when $\mu \leq 0.3$ compared with the PixelDP model. On average, our HGM_PixelDP ($\epsilon_r = 4$) improves 4.17% in terms of certified accuracy given $\mu_a \leq 0.3$, compared with the PixelDP, $p < 5.89e - 3$ (2 tail t-test). Given very strong adversarial perturbation $\mu_a > 0.3$, smaller ϵ_r usually yields better results, offering the flexibility in choosing appropriate DP budget ϵ_r for robustness given different attack magnitudes. These experimental results show crucial benefits of relaxing the constraints of the privacy budget and of the heterogeneous noise distribution in our HGM.

Secure-SGD. The application of our HGM in DP-preserving deep neural networks, i.e., Secure-SGD, further strengthens our observations. Figures 4 and 5 illustrate the certified accuracy under attacks of each model as a function of the privacy budget ϵ used to protect the training data. By incorporating HGM into DPSGD, our Secure-SGD remarkably increases the robustness of differentially private deep neural networks. In fact, our Secure-SGD with HGM outmatches DGSGP, AdLM, and the application of AGM in our Secure-SGD algorithm in most of the cases. Note that the application of AGM in our Secure-SGD does not redistribute the noise in deriving the provable robustness. In CIFAR-10 dataset, our Secure-SGD ($\epsilon_r = 8$) correspondingly acquires a 2.7% gain ($p < 1.22e - 6$, 2 tail t-test), a 3.8% gain ($p < 2.16e - 6$, 2 tail t-test), and a 17.75% gain ($p < 2.05e - 10$, 2 tail t-test) in terms of conventional accuracy, compared with AGM in Secure-SGD, DPSGD, and AdLM algorithms. We register the same phenomenon in the MNIST dataset. On average, our Secure-GSD ($\epsilon_r = 4$) correspondingly outperforms the AGM in Secure-SGD and DPSGD with an improvement of 2.9% ($p < 8.79e - 7$, 2 tail t-test) and an improvement of 10.74% ($p < 8.54e - 14$, 2 tail t-test).

Privacy preserving and provable robustness. We also discover an original, interesting, and crucial trade-off between DP preserving to protect the training data and the provable robustness (Figures 4 and 5). Given our Secure-SGD model, there is a huge improvement in terms of conventional accuracy when the privacy budget ϵ increases from 0.2 to 2 in MNIST dataset (i.e., 29.67% on average), and from 2 to 10 in CIFAR-10 dataset (i.e., 18.17% on average). This opens a long-term research avenue to achieve better provable robustness under strong privacy guarantees, since with strong privacy guarantees (i.e., small values of ϵ), the conventional accuracies of all models are still modest.

6 Conclusion

In this paper, we presented a Heterogeneous Gaussian Mechanism (HGM) to relax the privacy budget constraint, i.e., from $(0, 1]$ to $(0, \infty)$, and its heterogeneous noise bound. An original application of our HGM in DP-preserving mechanism with provable robustness was designed to enhance the robustness of DP deep neural networks, by introducing a novel Secure-SGD algorithm with a better robustness bound. Our model shows promising results and opens a long-term avenue to address the trade-off between DP preservation and provable robustness. In future work, we will learn how to identify and incorporate more practical Gaussian noise distributions to further improve the model accuracies under model attacks.

Acknowledgements

This work is partially supported by grants DTRA HDTRA1-14-1-0055, NSF CNS-1850094, NSF CNS-1747798, NSF IIS-1502273, and NJIT Seed Grant.

[2] https://github.com/haiphanNJIT/SecureSGD

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *arXiv:1607.00133*, 2016.

[Bach *et al.*, 2015] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015.

[Balle and Wang, 2018] Borja Balle and Yu-Xiang Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 394–403, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[Carlini and Wagner, 2017] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, May 2017.

[Chatzikokolakis *et al.*, 2013] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In Emiliano De Cristofaro and Matthew Wright, editors, *Privacy Enhancing Technologies*, pages 82–102, 2013.

[Dong *et al.*, 2017] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Discovering adversarial examples with momentum. *CoRR*, abs/1710.06081, 2017.

[Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Dwork *et al.*, 2006] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, pages 265–284, 2006.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 1322–1333, 2015.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[Hamm *et al.*, 2017] J. Hamm, J. Luken, and Y. Xie. Crowdml: A library for privacy-preserving machine learning on smart devices. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6394–6398, 2017.

[Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[Kolter and Wong, 2017] J. Zico Kolter and Eric Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. *CoRR*, abs/1711.00851, 2017.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[Lecun *et al.*, 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lecuyer *et al.*, 2018] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *arXiv:1802.03471*, 2018.

[Lee and Kifer, 2018] Jaewoo Lee and Daniel Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1656–1665, 2018.

[Liu *et al.*, 2016] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick D. McDaniel, Arunesh Sinha, and Michael P. Wellman. Towards the science of security and privacy in machine learning. *CoRR*, abs/1611.03814, 2016.

[Phan *et al.*, 2017] N. Phan, X. Wu, H. Hu, and D. Dou. Adaptive laplace mechanism: Differential privacy preservation in deep learning. In *IEEE ICDM'17*, 2017.

[Raghunathan *et al.*, 2018] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *CoRR*, abs/1801.09344, 2018.

[Shokri *et al.*, 2017] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, May 2017.

[Wang *et al.*, 2015] Yue Wang, Cheng Si, and Xintao Wu. Regression model fitting under differential privacy and model inversion attack. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1003–1009, 2015.

[Yu *et al.*, 2019] L. Yu, L. Liu, C. Pu, M. Gursoy, and S. Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 326–343, 2019.